Supplementary for: Efficient Hyperparameter Optimization for Deep Learning Algorithms Using Deterministic RBF Surrogates

Ilija Ilievski
Graduate School for Integrative Sciences and Engineering
National University of Singapore
ilija.ilievski@u.nus.edu

Taimoor Akhtar
Industrial and Systems Engineering
National University of Singapore
erita@nus.edu.sg

Jiashi Feng
Electrical and Computer Engineering
National University of Singapore
elefjia@nus.edu.sg

Christine Annette Shoemaker
Industrial and Systems Engineering
National University of Singapore
isesca@nus.edu.sg

Appendix A: Efficiency comparison

Mean best validation error

In this subsection, we show progress charts for each of the four problems (Figure 1, Figure 2, Figure 3, and Figure 4). The charts show mean best found solution per function evaluation, so we can observe how quickly each of the algorithms make progress towards finding better hyperparameter set. The mean is taken over 5 trials with different random seed for all problems, except for the first problem, where is taken over 10 trials.

Mean validation error

In this subsection, we show the mean validation error per function evaluation (Figure 5, Figure 6, Figure 7, and Figure 8). The mean is taken over 5 trials with different random seed for all problems, expcept for the first problem, where is taken over 10 trials. In this figures we can observe how each of the algorithms explore the hyperparameter space, and how they reached the best validation error per trial.

Appendix B: Experiments details

Multi-layer Perceptron Network

The MLP network consists of two hidden layers with *ReLU* activation between them and *SoftMax* at the end. As learning algorithm we use Stochastic Gradient Descent (SGD). We optimize the hyperparameters of the learning algorithm, the layer weight initialization hyperparameters, and network structure hyperparameters. The hyperparameters being optimized for this network are listed in Table 1. The default column shows the point used as initial starting point for SMAC.

Convolutional Neural Network

The CNN network consists of two convolutional blocks, each containing one convolutional layer with batch normalization, followed by *ReLU* activation and 3×3 max-pooling. Following the convolutional blocks, are two fully-connected layers with *LeakyReLU* activation, and *SoftMax* layer at the end.[1] The hyperparameters being optimizing for this network are listed in Table 2, Table 3, and Table 4. The default column shows the point used as initial starting point for SMAC and HORD-ISP.

MNIST

The MNIST dataset contains 60,000 training images and 10,000 testing images. Following conventional experimental protocol on this dataset, we split the training images into the training set of 50,000 images and the validation set of 10,000 images.

CIFAR-10

The CIFAR-10 dataset consists of 60,000 color images equally divided in 10 classes. The dataset is split into five training batches and one test batch, each with 10,000 images. We choose the last training batch as a validation set and use the error on this set to compare the performance of the algorithms.

Appendix C: Non-evaluation time

Here, we report the mean non-evaluation time – the time needed by each algorithm to propose point in the hyperparameter space (averaged across 5 trials). We report the median across function evaluations times for the 15 and 19 dimensional problems.

[1] LeakyReLU is defined as $f(x) = \max(0, x) + \alpha * \min(0, x)$, where α is a hyperparameter.

Table 1: Details for optimizing 6 hyperparameters of an MLP network used in first experiment and applied on MNIST (**6-MLP**).

Hyperparameter	Type	Range	Default (ISP)
Learning rate of SGD	Continuous	$[0.001 - 0.20]$	0.1
Momentum of SGD	Continuous	$[0.80 - 0.999]$	0.9
Mean of Gaussian initialization	Continuous	$[0.00 - 0.01]$	0.00
STD of Gaussian initialization	Continuous	$[0.001 - 1.00]$	0.01
Number of training epochs	Integer	$[8 - 20]$	10
Number of hidden nodes	Integer	$[50 - 200]$	50

Table 2: Details for optimizing 8 hyperparameters of a CNN used in second experiment and applied on MNIST (**8-CNN**).

Hyperparameter	Type	Range	Default (ISP)
Learning rate of SGD	Continuous	$[0.005 - 0.30]$	0.1
Momentum of SGD	Continuous	$[0.60 - 0.999]$	0.9
Weight decay rate	Continuous	$[0.00 - 0.01]$	0.0005
Learning rate decay	Continuous	$[0.001 - 1.00]$	0.0002
Number of hidden nodes in first Conv Layer	Integer	$[16 - 128]$	32
Number of hidden nodes in second Conv Layer	Integer	$[16 - 128]$	64
Number of hidden nodes in first FC Layer	Integer	$[100 - 400]$	200
Number of hidden nodes in second FC Layer	Integer	$[100 - 400]$	256

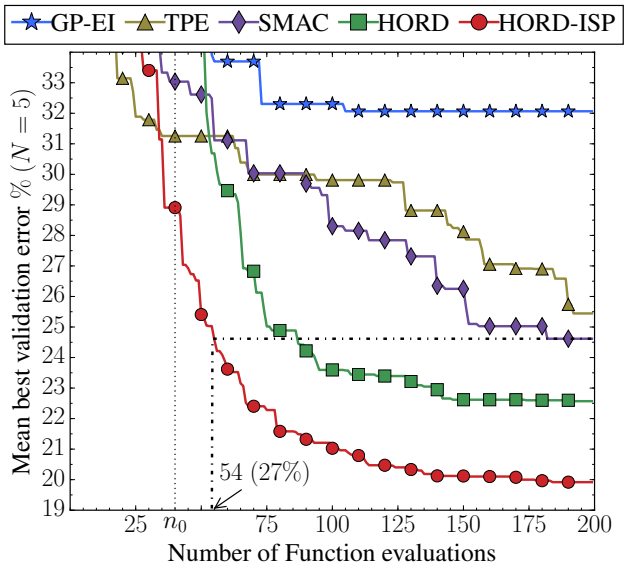

Figure 1: Efficiency comparison of HORD and HORD-ISP with baselines. The methods are used for optimizing a CNN with 19 hyperparameters on the CIFAR-10 dataset (**19-CNN**). We plot validation error curves of the compared methods against number of the function evaluations (averaged over 5 trials). HORD and HORD-ISP show to be significantly more efficient than other methods. HORD-ISP only takes 54 function evaluations to achieve the lowest validation error that the best baseline (SMAC) gives after 200 evaluations.

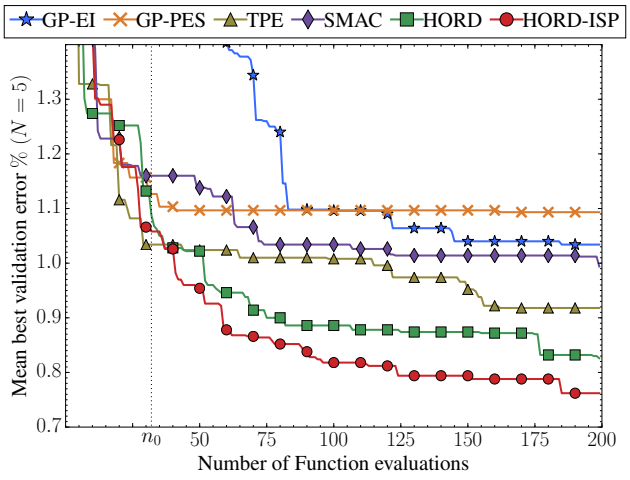

Figure 2: Efficiency comparison of HORD and HORD-ISP with baselines. The methods are used for optimizing a CNN with 15 hyperparameters on the MNIST dataset (**15-CNN**). We plot validation error curves of the compared methods against number of the function evaluations (averaged over 5 trials). HORD and HORD-ISP show to be significantly more efficient than other methods.

Table 3: Details for optimizing 15 hyperparameters of a CNN used in the third experiment and applied on MNIST (**15-CNN**).

Hyperparameter	Type	Range	Default (ISP)
Learning rate of SGD	Continuous	$[0.005 - 0.30]$	0.1
Momentum of SGD	Continuous	$[0.60 - 0.999]$	0.9
Weight decay rate	Continuous	$[0.00 - 0.01]$	0.0005
Learning rate decay	Continuous	$[0.001 - 1.00]$	0.0002
Alpha parameter of leaky *ReLU* in first FC layer	Continuous	$[0.00 - 0.50]$	0.01
Alpha parameter of leaky *ReLU* in second FC layer	Continuous	$[0.00 - 0.50]$	0.01
STD of Gaussian initialization for first FC layer	Continuous	$[0.00 - 0.50]$	0.01
STD of Gaussian initialization for second FC layer	Continuous	$[0.00 - 0.50]$	0.01
STD of Gaussian initialization for first Cong layer	Continuous	$[0.00 - 0.50]$	0.01
STD of Gaussian initialization for second Conv layer	Continuous	$[0.00 - 0.50]$	0.01
Mini-batch size	Integer	$[32 - 512]$	128
Number of hidden nodes in first Conv Layer	Integer	$[16 - 128]$	32
Number of hidden nodes in second Conv Layer	Integer	$[16 - 128]$	64
Number of hidden nodes in first FC Layer	Integer	$[100 - 400]$	200
Number of hidden nodes in second FC Layer	Integer	$[100 - 400]$	256

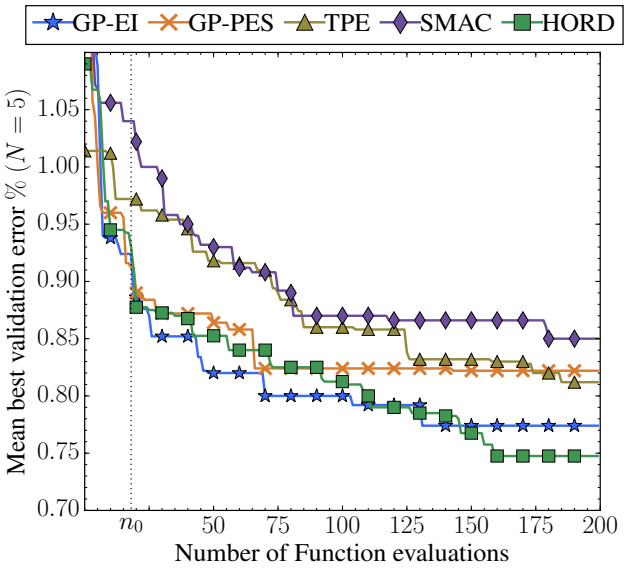

Figure 3: Efficiency comparison of HORD with baselines. The methods are used for optimizing a CNN with 8 hyperparameters on the MNIST dataset (**8-CNN**). We plot validation error curves of the compared methods against number of the function evaluations (averaged over 5 trials).

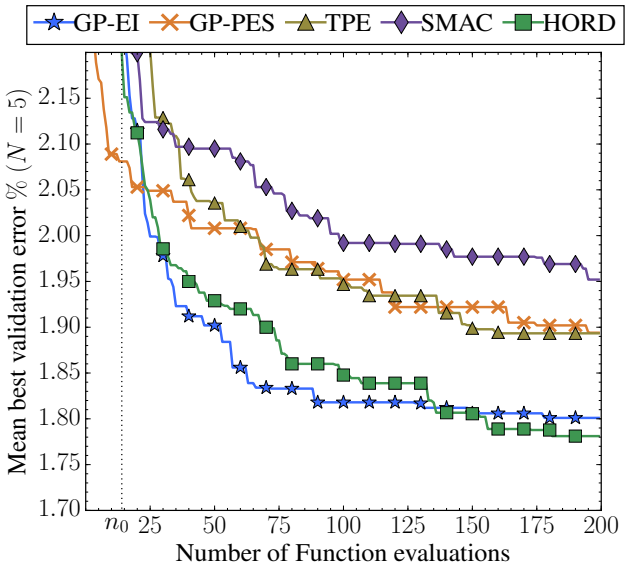

Figure 4: Efficiency comparison of HORD with baselines. The methods are used for optimizing an MLP network with 6 hyperparameters on the MNIST dataset (**6-MLP**). We plot validation error curves of the compared methods against number of the function evaluations (averaged over 10 trials).

Table 4: Details for optimizing 19 hyperparameters of a CNN used in the forth experiment and applied on CIFAR-10 (**19-CNN**).

Hyperparameter	Type	Range	Default (ISP)
Learning rate of SGD	Continuous	$[0.005 - 0.30]$	0.1
Momentum of SGD	Continuous	$[0.60 - 0.999]$	0.9
Weight decay rate	Continuous	$[0.00 - 0.01]$	0.0005
Learning rate decay	Continuous	$[0.001 - 1.00]$	0.0002
Alpha parameter of leaky *ReLU* in first FC layer	Continuous	$[0.00 - 0.50]$	0.01
Alpha parameter of leaky *ReLU* in second FC layer	Continuous	$[0.00 - 0.50]$	0.01
STD of Gaussian initialization for first FC layer	Continuous	$[0.00 - 0.50]$	0.01
STD of Gaussian initialization for second FC layer	Continuous	$[0.00 - 0.50]$	0.01
STD of Gaussian initialization for first Conv layer	Continuous	$[0.00 - 0.50]$	0.01
STD of Gaussian initialization for second Conv layer	Continuous	$[0.00 - 0.50]$	0.01
Dropout rate for first FC layer	Continuous	$[0.00 - 0.80]$	0.5
Dropout rate for second FC layer	Continuous	$[0.00 - 0.80]$	0.5
Dropout rate for first Conv layer	Continuous	$[0.00 - 0.80]$	0.5
Dropout rate for second Conv layer	Continuous	$[0.00 - 0.80]$	0.5
Mini-batch size	Integer	$[32 - 512]$	128
Number of hidden nodes in first Conv Layer	Integer	$[16 - 128]$	32
Number of hidden nodes in second Conv Layer	Integer	$[16 - 128]$	64
Number of hidden nodes in first FC Layer	Integer	$[100 - 400]$	200
Number of hidden nodes in second FC Layer	Integer	$[100 - 400]$	256

Table 5: Median non-evaluation time of each algorithm for optimizing 15 and 19 hyperparameters of a CNN on MNIST.

Algorithm	Non-evaluation time for **15-CNN** in seconds	Non-evaluation for **19-CNN** in seconds
GP-EI	69.4s	318.72s
GP-PES	430.7s	–
TPE	0.06s	0.086s
SMAC	0.385s	0.837s
HORD	0.04s	0.06s
HORD-ISP	0.04s	0.06s

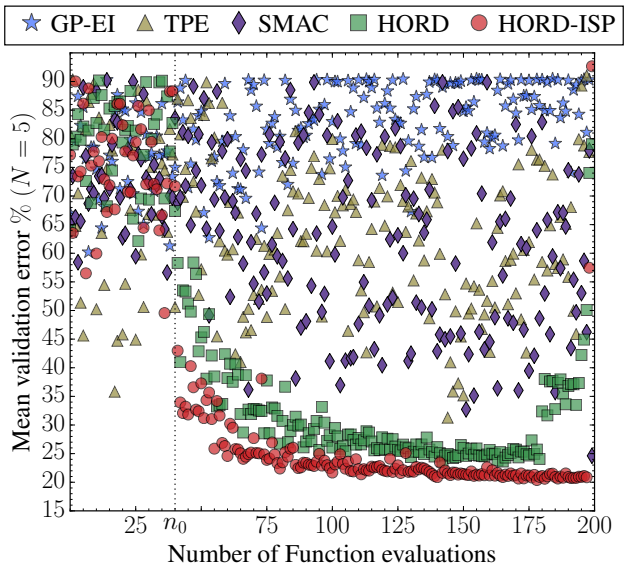

Figure 5: Mean validation error *v.s.* number of function evaluations of different methods for optimizing 19 hyperparameters of CNN (**19-CNN**) on CIFAR-10. One dot represents validation error of an algorithm at the corresponding evaluation instance. After n_0 evaluations, the searching of HORD and HORD-ISP starts to focus on the hyperparameters with smaller validation error ($\leq 35\%$), in stark contrast with other methods.

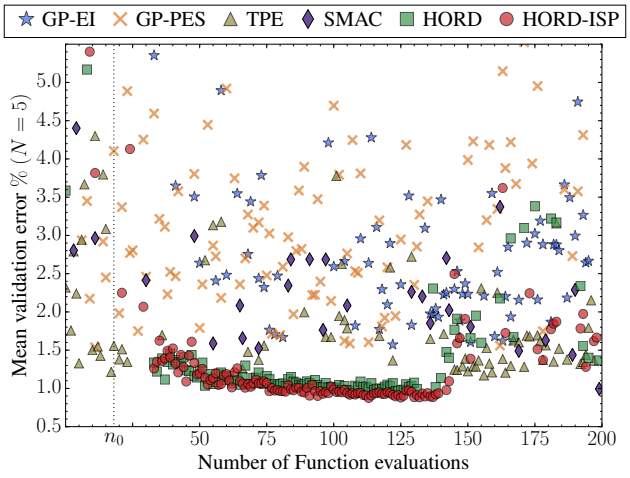

Figure 6: Mean validation error *v.s.* number of function evaluations of different methods for optimizing 15 hyperparameters of CNN (**15-CNN**) on MNIST (averaged over 5 trials). One dot represents validation error of an algorithm at the corresponding evaluation instance.

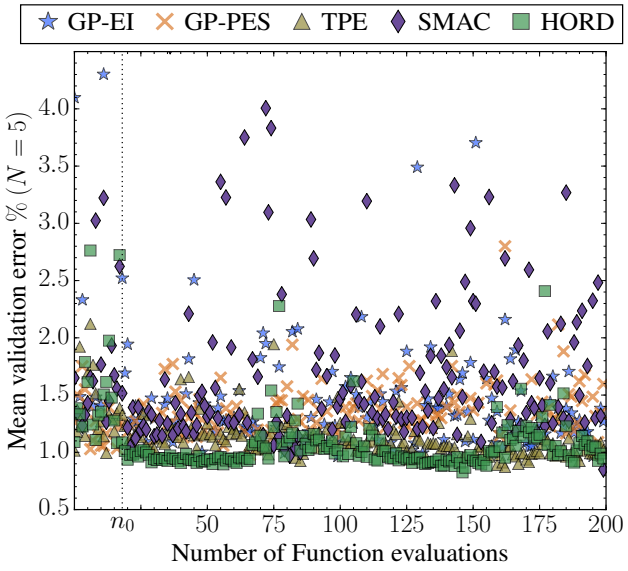

Figure 7: Mean validation error *v.s.* number of function evaluations of different methods for optimizing 8 hyperparameters of CNN (**8-CNN**) on MNIST (averaged over 5 trials). One dot represents validation error of an algorithm at the corresponding evaluation instance.

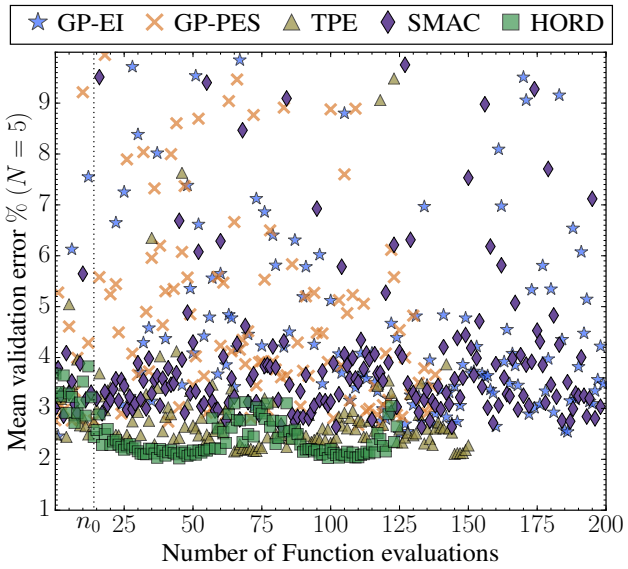

Figure 8: Mean validation error *v.s.* number of function evaluations of different methods for optimizing 6 hyperparameters of MLP (**6-MLP**) network on MNIST (averaged over 10 trials). One dot represents validation error of an algorithm at the corresponding evaluation instance.

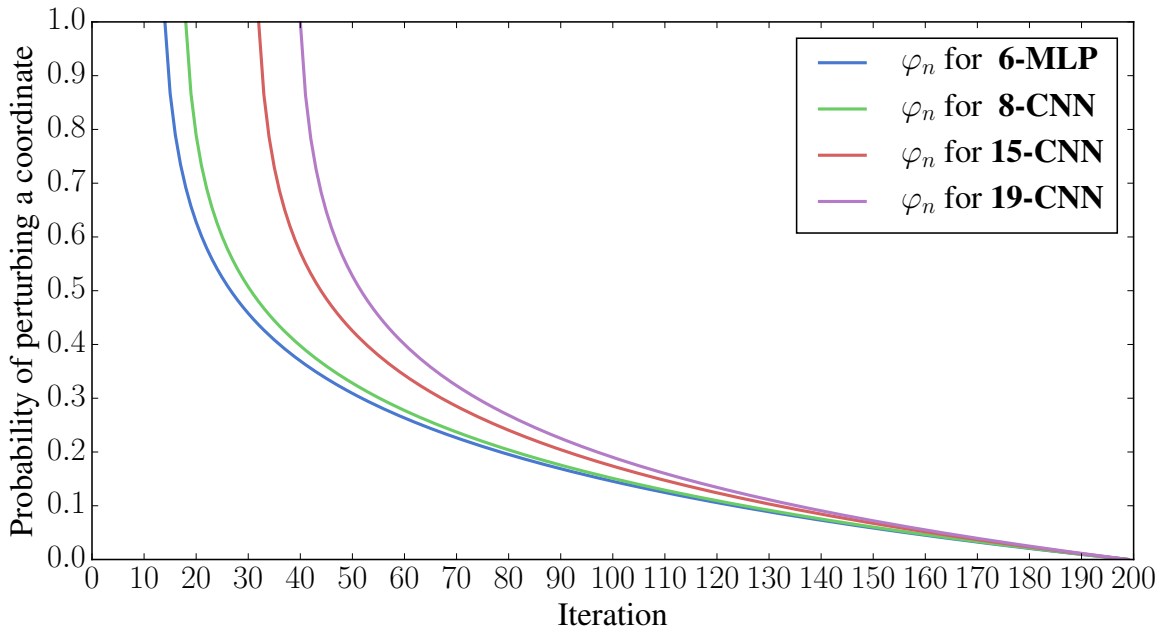

Figure 9: Probability of perturbing a coordinate φ_n per iteration, for each of the four experiments. The probability is given by $\varphi_n = \varphi_0 \left[1 - \frac{\ln(n-n_0+1)}{\ln(N_{\max}-n_0)}\right], \quad n_0 <= n < N_{\max}$, where $\varphi_0 = \min(20/D, 1)$, D is the number of hyperparameters being optimized, n is the iteration number, and N_{max} is the maximum number of allowed iterations.